

iranphp articles

عنوان مقاله : پیاده سازی رویداد گرایی در php
نگارنده : میلاد راستیان
آدرس پست الکترونیک : miladmovie@yahoo.com
تاریخ نگارش :

پیاده سازی رویداد گرایی در php :

ما می خواهیم در مورد روش پیدا سازی رویداد گرایی (event-driven) در php بحث کنیم .
 من خودم خیلی وقت بود که تقریباً از چنین روشی استفاده می کردم. ولی چندان جالب نبود تا با خواندن کتاب Professional PHP5 یک سری ایده به ذهنم خورد که آنان در موردش صحبت می کنیم البته باید بگم این کد در PHP4 هم کاملاً جواب داده.
 خب خود شما از چه روشی استفاده می کنید ؟ فکر کنم به صورت زیر :

old_script.php

```
<html>
<head>
<title>You doing</title>
</head>
<body>
<?= submit() ?>
<form method="post" action="old_script.php">
<input type="text" name="name" />
<input type="hidden" name="formsubmit" value="1" />
<input type="submit" name="btnsubmit" value="Submit It ! " />
</form>
</body>
</html>
<?php
Function submit()
{
    if(isset($_POST['formsubmit']))
    {
        return "form submitted and name value is : ".$_POST['name'];
    }
}
?>
```

خب البته این هم روشی واسه خودش ولی فکر کنید هر بار فرمی می خواهید درست کنید باید این همه زحمت به خودتون بدید ! حالا فکر کنید توی فرم دوتا دکمه باشه که هر کدوم یک کار رو انجام بدن . خب اینجاست که کلاس های و این شیوه رویداد گرایی به کمک من و شما میانه !

handler.php

```
<?php
function handler ($classname,$method='POST')
{
    if(class_exists($classname))
    {
        if($_SERVER['REQUEST_METHOD']==$method)
        {
            if(isset($_POST['handle_event']))
            {
                $obj=new $classname ();
                $name="handler_".$_POST['handle_event'];
                $obj->{$name} ();
            }
        }
    }
    else
        echo "can not handle this event !";
}
?>
```

تابع دو ورودی دارد نام کلاس و نوع متدی که می خواهیم برای آن رویداد بنویسیم (POST or GET) چک می کند که آیا چنین کلاسی وجود دارد اگر صفحه به صورت متدی که مورد نظر ما است لود شده باشد .

در صورتی که نیازی به رویداد گرای برای این صفحه باشد
یک شی از این کلاس درست می شود و تابع که با رویداد مورد نظر مطابقت داشته باشد اجرا می شود

test.class.php

```
<?php
class test
{
    function handler_show()
    {
        echo 'form is submitted and name is <b>'.$_POST['name'].'</b>';
    }
}
?>
```

new_script.php

```
<?php
require("test.class.php");
require("handler.php");
?>
<html><head><title>You doing</title></head>
<body>
<?php
handler ('test');
?>
<form method="post" action="new_script.php">
<input type="text" name="name" />
<input type="hidden" name="handle_event" value="show" />
<input type="submit" name="btnsubmit" value="Submit It !" />
</form>
</body>
</html>
```

در این فرم یک فیلد مخفی است که نشان می دهد چه عملی را می خواهیم انجام بدهیم و در کدها handler_ را به اول این عمل اضافه می کنیم و یک تابع به کلاس با این نام اضافه می کنیم
فرض کنیم در صورتی که بخواهیم نام را از فرم بگیرد و به صورت بزرگ نشان دهد تنهای مقدار فیلد مخفی handle_event را به show_big تغییر می دهیم و یک تابع به نام handler_show_big مانند زیر به کلاس مان اضافه می کنیم :

```
function handler_show_big()
{
    echo '<h1>'.$_POST[' name ']. '</h1>';
}
```

این مثالی که من در اینجا آوردم شاید خیلی ساده به نظر بیاد ولی وقتی پروژه تون اون قدر بزرگ شد که ممکنه جایی تابع های رو که استفاده کردید یادتون بره مطمئنا با این روش به خیلی راحت می تونید تابع ها رو ویرایش کنید.